**Matplotlib**

Matplotlib เป็นไลบรารี่สำหรับการพล็อตโดยมีภาษาไพธอนเป็นฐาน ( Python-based plotting library ) ด้วยกับการสนับสนุนอย่างเต็มที่สำหรับการพล็อตในรูปแบบสองมิติ ( 2D ) อีกทั้งรองรับสำหรับการพล็อตกราฟฟิกสามมิติ ( 3D graphics ) โดยใช้กันอย่างแพร่หลายในแวดวงของการคำนวณทางวิทยาศาสตร์ด้วยภาษาไพธอน เป้าหมายของไลบรารี่ Matplotlib มุ่งเป้าไปที่กรณีการใช้งานที่หลากหลาย มันสามารถที่จะฝั่งกราฟฟิกในชุดเครื่องมือการติดต่อกับผู้ใช้งาน ( user interface toolkit ) และตอนนี้รองรับกราฟฟิกที่สามารถตอบสนองได้บนระบบปฏิบัติการที่ใช้กันอย่างแพร่หลายด้วยชุดเครื่องมือ GTK+, Qt, Tk, FLTK, wxWidgets และ Cocoa มันสามารถที่จะเรียกใช้อย่างโต้ตอบผ่าน interactive Python shell เพื่อสร้างภาพกราฟฟิกด้วยคำสั่งขั้นตอนอย่างง่ายเช่น Mathematica, IDL และ MATLAB และมันสามารถที่จะฝั่งตัวใน headless webserver เพื่อจัดเตรียมฉบับพิมพ์ ( hardcopy ) ในรูปแบบ raster-based เช่น Portable Network Graphics ( PNG ) และรูปแบบเวกเตอร์เช่น PostScript และรูปแบบ Portable Document Format ( PDF ) และรูปแบบสุดท้าย Scalable Vector Graphics ( SVG )

**วัตถุประสงค์**

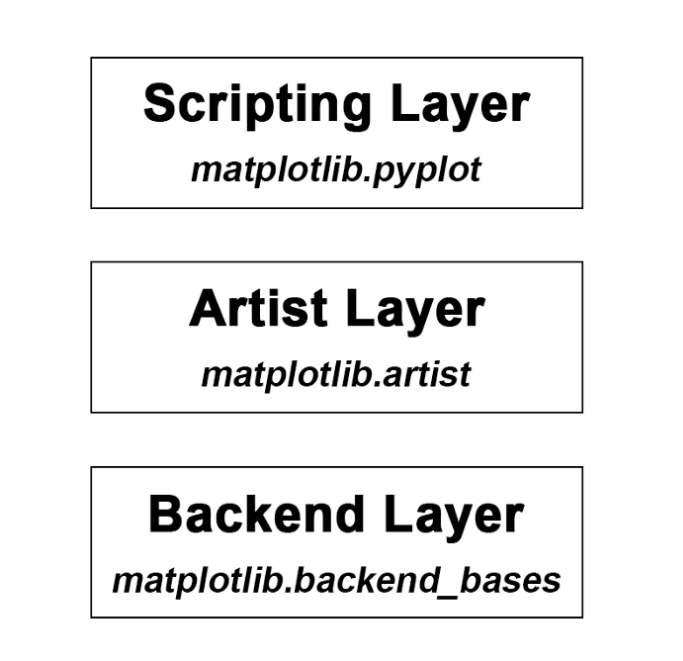
Matplotlib เป็นไลบรารี่ที่ครอบครุมสำหรับการแสดงภาพ ( visualization ) ในรูปแบบคงที่ ( static ) เคลื่อนไหว ( animated ) โต้ตอบได้ ( interactive ) ด้วยภาษาไพธอน Matplotlib ช่วยให้ทำสิ่งเหล่านี้ให้ง่ายขึ้นและสะดวกยิ่งขึ้น

**Architectural Patterns / Styles**

รูปแบบสถาปัตยกรรมที่ Matplotlib ใช้เป็นรูปแบบ Layer architectural โดยมีทั้งหมดสามเลเยอร์

* Scripting Layer เป็นเลเยอร์ที่มีการเขียนสคริปต์ที่เบาที่สุด ( Scripting interface ) ในบรรดาเลเยอร์ทั้งสาม ถูกออกแบบมาเพื่อทำให้ไลบรารี่ Matplotlib ทำงานคล้ายคลึงกับ MATLAB script เป็นเลเยอร์ที่อยู่ชั้นบนสุด เป็นเลเยอร์ที่รวบรวมชุดคำสั่งและง่ายต่อการใช้งาน Artist Layer เป็นเลเยอร์ที่หนักในเชิงวากยสัมพันธ์ ( syntactic ) เนื่องจากมันถูกออกแบบว่าไว้สำหนับนักพัฒนาและไม่ได้ออกแบบมาไว้สำหรับคนที่ต้องการศึกษาอย่างรวบรัดดังนั้น Scripting Layer เป็นเลเยอร์ที่ง่ายต่อการใช้งาน บางครั้งอาจจะเรียกว่า procedural plotting
* Artist Layer เป็นเลเยอร์ที่ช่วยให้ควบคุมและปรับจูนของ figure ให้ได้มากที่สุดเท่า เปรียบดั่งนักจิตรกรที่กำลังวาดบนผ้าใบ เลเยอร์นี้ประกอบไปด้วยหนึ่งออปเจ็คหลักคือ Artist ที่ใช้ Renderer เพื่อวาดภาพบนแคนวาส มันสามารถที่จะปรับแต่งได้มากว่าเมื่อเปรียบเทียบกับ Scripting Layer และมันสะดวกกว่าสำหรับการพล็อตขั้นสูง โดยเฉพาะอย่างยิ่งเมื่อจัดการเกี่ยวกับหลายรูปภาพหรือหลายแกนมันจะทำให้เราไม่สับสนว่าเรากำลังทำงานอยู่กับภาพหรือแกนไหนเนื่องจากทุก ๆ subplot ทั้งหมดถูกกำหนดค่าไปยังออปเจ็คของ Artist ทำให้บางครั้งสามารถที่เรียก Artist Layer ว่า object-based plotting ทุก ๆ สิ่งที่เห็นบน Matplotlib figure เป็นอินสแตนท์ของ artist เช่น Title, lines, tick labels, images เป็นต้น ออปเจ็คของ Artist มีอยู่ 2 ประเภท ได้แก่ primitive type เช่น line2d, rectangle, circle และ text รูปแบบที่สองคือ composite type เช่น Axis, Tick, Axes และ Figure
* Backend Layer เป็นเลเยอร์ที่จัดการงานที่หนักผ่านการสื่อสารไปยังชุดเครื่องมือวาดภาพในเครื่องเช่น wxPython หรือภาษาวาดภาพ ( drawing language ) อย่าง PostScript เป็นเลเยอร์ที่ซับซ้อนที่สุด ประกอบไปด้วยสามบิลท์อินคลาสหลัก ๆ ( built-in abstract interface classes ) 1. FigureCanvas เป็นแคสวาสที่จะแสดงรูปภาพ 2. Renderer เป็น abstract class ที่จัดการในเรื่องการวาดและการแสดงผล มีหน้าที่ในการวาด ใน FigureCanvas 3. Event จัดการเกี่ยวการป้อนของผู้ใช้งานเช่น การกดคีย์บอร์ดและการกดเมาส์เป็นต้น

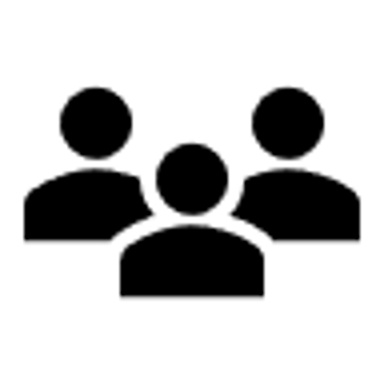
รูปภาพประกอบเพื่อให้เห็นภาพ



รูป 1 <https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569>

**Quality Attribute Scenarios**

Modifiability



Source:

Developer

Stimulus:

Wishes to Change Code

Artifact:

Code

Environment:

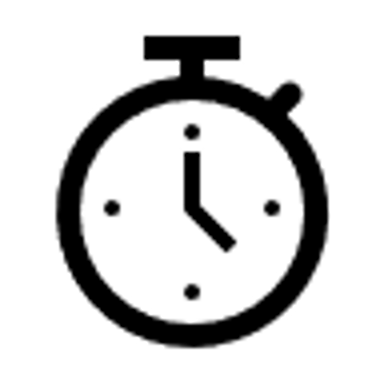
Development Time

Response:

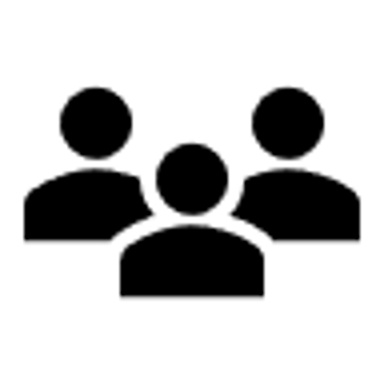
Modification is made with no side effects

Response measure:

In 3 hours



Portability



Source:

OS

Stimulus:

Want to run on another OS

Artifact:

Resource

Environment:

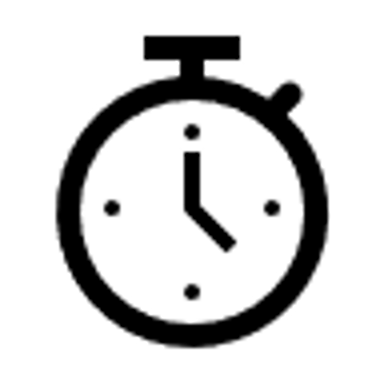
Runtime

Response:

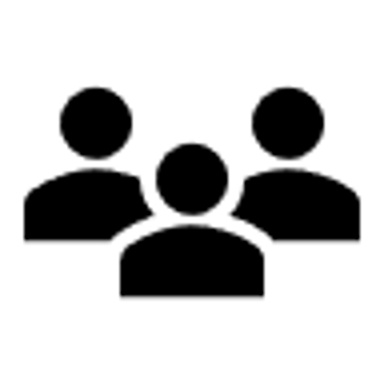
Can run without error occurs

Response measure:

In 1 hour



Testability



Source:

Developer

Stimulus:

Complete a code unit

Artifact:

Code Unit

Environment:

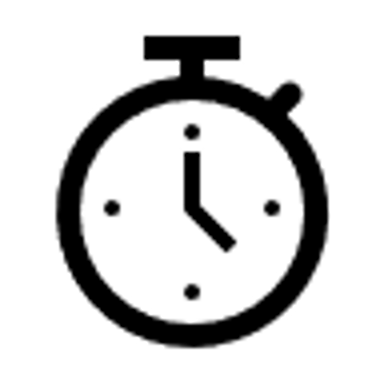
Development

Response:

Perform a test sequence

Response measure:

85% path coverage in 20 minutes



**แหล่งอ้างอิง**

* <https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569>
* <https://www.aosabook.org/en/matplotlib.html>